MRLPLT USER'S GUIDE – A GENERAL PURPOSE PROGRAM FOR GRAPHICAL REPRESENTATION OF DATA OVER 2-D REGIONS (VAX-11/750 VERSION)

Y.S. Yu, N.A. Toews, A.S. Wong and H.D. Morrison

MRL 88-97(TR)

MRLPLT USER'S GUIDE - A GENERAL PURPOSE PROGRAM FOR GRAPHICAL REPRESENTATION OF DATA OVER 2-D REGIONS (VAX-11/750 VERSION)

by

Y. S. Yu* N.A. Toews*, A.S. Wong** and H. D. Morrison***

ABSTRACT

MRLPLT is a general purpose plotting program designed for finite element analysis of two dimensional structures. It is used together with the interface program *SAPPLT* and functions as the post-processor for the *SAP2D* system.

MRLPLT was originally developed and tested for a CDC computer and the Calcomp plotters. It has been used, at the Mining Research Laboratories (MRL), for a number of years, and, has proved to be very useful and flexible. Recently, a number of modifications to *MRLPLT*, were carried out, and, *MRLPLT* is operational on MRL's DIGITAL VAX-11/750 mini-computer under VMS operating system. The plotting devices associated with this conversion are, now, restricted to a Tektronix PLOT10 compatible graphic display terminals and the Calcomp 1042 ploter.

MRLPLT provides graphical representation of stresses and displacements or other quantities resulting from finite element analysis of two dimensional structures. The three types of plots available in this program are vectors, tensors, and point contours.

This report documents *MRLPLT*. In particular, the files used, the plotting capabilities and options, and input instructions are described. In addition, a command procedure for VAX/VMS operating system is provided, and, plotting examples are illustrated.

* Research Scientists, ** Physical Scientist, Mining Research Laboratories, CANMET, Energy, Mines and Resources Canada, Ottawa. *** Formerly a Co-op student, University of Waterloo.

GUIDE D'UTILISATION DE MRLPLT, PROGRAMME UNIVERSEL DE REPRÉSENTATION GRAHPIQUE DE DONNÉES SUR DES STRUCTURES BIDIMENSIONNELLES (VERSION VAX-11/750)

par

Y. S. Yu* N.A. Toews*, A.S. Wong** and H. D. Morrison***

RÉSUMÉ

MRLPLT est un programme universel de traçage conçu pour l'analyse par éléments finis de structures bidimensionnelles. Il est utilisé conjointement avec le programme d'interfaçage *SAPPLT* et agit comme post-processeur pour le système *SAP2D*.

MRLPLT a à l'origine été mis au point et à l'essai pour un ordinateur CDC et les traçeurs Calcomp. Il a été utilisé aux Laboratoires de recherche minière (LRM) pendant plusieurs années et s'est avéré très utile et très souple. Récemment, plusierurs modifications ont été apportées à *MRLPLT* qui est désormais exploitable sur le mini-ordinateur VAX–11/750 DIGITAL des LRM par l'intermédiaire du système d'exploitation VMS. Les dispositifs de traçage associés à cette conversion sont maintenant limités à un PLOT10 Tektronix, terminal d'affichage graphique compatible, et au traçeur Calcomp 1042.

MRLPLT réalise une représentation graphie des contraintes et des déplacements ou d'autres paraméetres résultant de l'analyse par éléments finis de structures bidimensionnelles. Les trois éléments représentés par ce programme sont les vecteurs, les tenseurs et les courbes de niveau en pointillé.

Le présent rapport documente *MRLPLT*, notamment les fichiers utilisés, les possibilités et les options de traçage, et les instructions pour l'entrée des données. En outre, une procédure de commande du système d'exploitation VAX/VMS est décrite, et des exemple de traçage sont donnés.

* Chercheurs scientifiques, ** Chercheur en sciences physiques, Laboratoires de recherche minière, CANMET, Énergie, Mines et Ressources Canada, Ottawa.

*** Anciennement étudiant stagiaire de l'université de Waterloo.

CONTENTS

	page
ABSTRACT	i
RÉSUMÉ	ii
1.0 INTRODUCTION	1
2.0 GENERAL DESCRIPTION OF *MRLPLT* PROGRAM	2
3.0 FILE USAGE	3
4.0 INPUT INSTRUCTIONS	3
4.1 General Input Description	3
4.2 Plot Data Files	3
4.2.1 Plot Data Format	4
4.2.2 Plot Data File Creation	4
4.3 Boundary Files	5
4.3.1 Bounadry File Format	5
4.3.2 Boundary Data File Creation	6
4.4 Plot Commands file	6
4.4.1 General Description	6
4.4.2 Plot Initial Commands	8
4.4.3 Plot Keyword Commands	9
4.4.4 Plot Parameter Commands	10
4.4.5 Error Checking	15
4.5 Interactive SYS$COMMAND Data Input	16
4.5.1 Plot Legend Parameters	16
5.0 OUTPUT	18
5.1 Printer Output	18
5.2 Plotter Output	19
5.3 Plotting Examples	19
6.0 *MRLPLT* COMMAND PROCEDURE	25
7.0 REFERENCES	26
APPENDIX	27

TABLES

1. An example of *MRLPLT* Command Input Data	20
2. An example of the Boundary Input Data	21

- Contents continued -

FIGURES

1. Tensor Plot - Stress trajectories 22
2. Vector Plot - Excavation Displacements 23
3. Point Contour Plot - Major Principal stresses 24

1.0 INTRODUCTION

Structural analysis using the finite element technique always involve massive output. To speed up interpretation of the finite element results a post processor which provides graphical representation of stresses, displacements or other quantities is desired.

This report documents the computer program *MRLPLT* which was developed to interpret results of any two-dimensional finite element analysis by providing graphical representation of stresses, displacements and/or other quantities. Three types of plots - *vectors*, *tensors* and *point contours* are available, each providing a specific symbolic representation of certain data.

The program is very easy to use because the input procedure has been made very flexible. In addition, extensive error checking with informative diagnostic messages has been provided for most of the input.

MRLPLT was originally developed and tested on a CDC CYBER 74 computer with a Calcomp drum plotter [1]. A number of modifications to *MRLPLT*, in accordance with Fortran 77 standard, were carried out, and it is operational on a VAX-11/750 minicomputer under VMS operating system at the Mining Research Laboratories (MRL). The plotting devices associated with this conversion are a Tektronix PLOT10 compatible graphic display terminal and a Calcomp 1042 plotter.

In addition, due to graphic software dependent, two separate executable files *i.e.*, a PLOT10 and a Calcomp version, are currently being maintained. The graphic software used in *MRLPLT* is compatible with either PLOT10 or Calcomp software library.

The version of *MRLPLT* using PLOT10 software can be run on a Tektronix 4014 or PLOT10 compatible graphic display terminal for a pre-review of the plots. The plot files are saved and a report quality plot can be produced directly on an **IMAGEN** 8/300 or **OmniLaser 2115** laser printer. The procedure to produce report quality plot on the laser printers have been described in detail [2,3].

2.0 GENERAL DESCRIPTION OF *MRLPLT* PROGRAM

This section will provide a brief description of the terminolory and main features of the plotting program. The detailed explanation will be given in Section 4.

The function of this program is to provide graphical representation of various quantities obtained from a finite element analysis of two dimensional structures using the *SAP2D* program [4], or other finite element programs. Any number of plots can be carried out. A plot consists of a plot label or title, a plot legend , a boundary and the symbolic representation of the data, called data points or plotting data. This program provides three types of plots, namely:

(a) *vectors*;

(b) *tensors*; and

(c) *point contours*.

The *vector* plot was designed to illustrate displacements or any other similar data. The *tensor* plot produces a representation of the two principal values of second-order tensors, such as the stress trajectories of the principal stresses for a plane structures. The *point contour* can be used for any quantity dependent on two independent variables, X and Y such as the two principal stresses in a two dimensional stress analysis; it assigns a symbol to each data point according the range of the data in relation to a set of contour intervals. The symbols used for each type of plot are illustrated in the example plots given in Section 5.

Each plot must have a boundary which is plotted as a set of straight lines. The boundary is determined by an ordered set of of boundary points and the type of lines to draw between them. The data points and boundary points of all plots are retained on files on storage devices such as a disk. These files can be created before or during the running this program.

The **plot commands** procedure of *MRLPLT* is very flexible and can be user-defined within certain restrictions. These commands are mainly labelled for easy recognition during the data preparation. Extensive error-checking with informative diagnostic messages are provided for most of the input.

The other features of the program are:

- Many of the **plot commands** input are optional.
- The ability to choose any area (rectangular) of the data points for plotting. This allow the user to magnify any area of the data to be plotted.

3.0 FILE USAGE

Five files are used in *MRLPLT*. These are as follows:–

Logical File Name	Description
PLTIN	The *MRLPLT* **plot data** file;
PLTBND	The **boundary** data file;
PLTCMD	The *MRLPLT* **plot commands** file;
FOR005	Interactive SYS$COMMAND data input;
FOR006	The *MRLPLT* print-output file.

4.0 INPUT INSTRUCTIONS

4.1 General input Description

The input to *MRLPLT* is divided into four (4) groups:

(a) A **plot data** file containing the plotting data points to be plotted;

(b) A **boundary** file containing a set of boundaries which define the area within which the data points will be plotted; and

(c) The **plot commands** determine how the **plot data** and **boundary** files will be processed. The **plot data** and **boundary** files may be used by more than one plot. However, each plot requires its own set of **plot commands**.

(d) Interactive SYS$COMMAND data input for **plot legend** parameters and **pause** command on Calcomp 1042 plotter to set plot origin and activated the plotter to start plotting.

4.2 **Plot Data** Files

A **plot data** file is a sequence of any number of data records followed by one 'end-of-file' (< *CTRL/Z* > for VMS) record. The number and the type of data items on a data record vary with the type of plot to be done. However, the first two items are always the global co-ordinates of the initial points from which the symbolic representation of the remaining items is to be plotted. These co-ordinates may be either in cartesian form, (X,Y), or in polar co-ordinate system, (R,ALPHA), where R is the radius and ALPHA is the angle.

A data record for a *vector* plot has four (4) items. In cartesian form they are X, Y, DX, DY and in polar form they become R1, ALPHA1, R2, ALPHA2. The last two items define the terminal point of the vector in local co-ordinates, *i.e.*, the point (X,

Y) or (R1, ALPHA1) is the origin and the the initial point of the vector. Note that DX and DY are the nodal displacement components in the x- and y-direction of the cartesian co-ordinates respectively.

There are five items in a data record for a *tensor* plot. The choice of a co-ordinate system does not affect the last three items which are SMAJ, SMIN, and PA in that order. SMAJ and SMIN are the major and minor principal values of a second-order tensor, *i.e.*, major and minor principal stresses; PA is the angle of inclination of SMAJ with respect to the x-axis.

For a *point contour* plot data record, it consists of three items. The first two items are the global co-ordinates, and the third item, Z, can be any data which the user would like 'contoured', *i.e.*, classified into specified ranges of values to be represented on the plot by alphanumeric symbols.

4.2.1 **Plot Data** Format

The default format for reading the data records of any **plot data** file is:

(1X, 5E15.7)

However, there is a plot parameter command which allows the user to substitute his own format code for any one file at a time. The detail will be given in Section 4.4.4(ii).

4.2.2 **Plot Data** File Creation

All the **plot data** files for any one plot job are stored together on an auxiliary, sequential I/O device such as a disk, during program execution. In the program this device is logical unit number 1 with mnemonic name LDATA. For external reference it has been given the logical file name **PLTIN**.

The **plot data** files may be placed on **PLTIN** in one of the two ways. One method is to create **PLTIN** by running an interface program before executing the program *MRLPLT*. The interface program will take the output file from a 2-D finite element program and create an file which can be used directly by *MRLPLT*. Two interface programs for two separate finite element programs, have been developed for this purpose at MRL; they are described in separate reports [5,6]. The alternative is to input the data records through the regular input stream, *i.e.*, one data record per line. Obviously, the latter is very tedious and is not recommended.

In both cases, the plot initial commands, $DATASET, must be input to signal the program which method is to be used (see Section 4.4.2(ii)). The **plot data** files are

referenced in the program by numbers corresponding to their position on **PLTIN**; the first file (for the first plot) is number 1, the second is 2, and so on.

4.3 **Boundary** Files

A **boundary** file is used to define the appearance of a region within which the data points are to be plotted. It is a sequence of any number of data records followed by one 'end-of-file' record. Each data record contains the items X, Y and NCODE, in that order. X and Y are the global co-ordinates of a boundary point and NCODE is a plotter pen-code. The boundary is plotted by straight line movements of the plotter pen from one boundary point to the next in the order they appear on the file. The type of the line drawn between points depends on the pen-code of the point to which the pen is moving. The pen-code, NCODE, may have one of the following three values:

(a) **2** (pen down) for a solid line;

(b) **3** (pen up) for no line; and

(c) **4** (pen down) for a dashed line.

If NCODE is not specified it is set to 2.

A boundary is not restricted to a simple geometry connected by a few straight lines. Therefore, a boundary is further defined as a set of one or more polygons, called loops. It is limited to **10** loops in the present program but it can easily be modified. On a **boundary** file a loop has a well-defined structure. It is an ordered set of data records such that the first and the last records have the same global co-ordinates. In addition, the first record must have NCODE set to 3 so that no line is drawn from the previous pen position to the starting point of the loop. The order of the records must be such that the data points, for which the boundary loop is being drawn, appear to the left the direction of movement of the plotter pen. In other words, if the interior of the loop is to contain data points then the order of the boundary points should describe the loop in a counter-clockwise direction. If the data points are to be plotted exterior to the loop then the order of the boundary points should describe the loop in a clockwise direction.

4.3.1 **Boundary** File Format

The default format for reading the data records of a **boundary** file is:

(1X, 2E15.7, I5)

However, there is a plot parameter command, $BNDRYFMT, which allows the user to substitute his own format code for any one file at a time. This plot parameter command is described in Section 4.4.4(iv).

4.3.2 **Boundary** Data File Creation

All the **boundary** files for one plot job are stored together on an auxiliary, sequential I/O device, such as a disk or tape, during program execution. In the program this device is logical unit number 2 with mnemonic name **LBOND**. For external reference it has been given the logical name **PLTBND**.

The **boundary** file may be placed on **PLTBND** in one of the two ways. One method to create **PLTBND** is to run another program, *BNDGEN*, using an output file generated from the finite element program *SAP2D*. The use of *BNDGEN* is described in Reference 5. The alternative is to input the data records with the regular input stream of this program (one data record per record or line). In both cases a plot initial command, $BNDRYSET, must be input to signal the program as to which method is to be used. The $BNDRYSET command is described in Section 4.4.2(ii). The files are referenced in the program by numbers corresponding to their position on **PLTBND**; the first file is number 1, the second is 2 and so on.

There is another way to input a **boundary** file for a plot. This is accomplished by using one of the plot parameter command - $BNDRY. The use of $BNDRY is described in Section 4.4.4 (iii). In this case, the type of **boundary** file is only stored for the duration of a plot on a temporary file.

4.4 **Plot Commands**

4.4.1 General Description

All the **plot commands** input by this program are either labelled or unlabelled. The labelled ones provide instructions to the program and data for the plot job. The unlabelled are used with certain labelled ones to input blocks of data such as the **plot data** and **boundary** files.

The **plot commands** files for one plot job are stored together on an auxiliary, sequential I/O device, such as a disk or tape, during program execution. In the program this device is logical unit number 7 with mnemonic name **LCR**. For external reference it has been given the logical name **PLTCMD**.

They are input with the character format, (A70), and copied to certain temporary files until the data they contain is required. More details concerning the use of these commands will follow.

All the commands (labelled) contain a nine character mnemonic label which appears in the first nine (9) columns of each of these commands. The tenth column is left

blank and the remaining 70 columns are for data items. Each **plot command** has a specific format which will be described individually. For simplicity, the label and blank column (10th column) are represented as one format item in the format lists as (A10), instead of their true input format which is (A1, 2A4, 1X). It should be noted that any label which does not appear as nine characters in its individual description should be padded on the right with blanks to extend it to nine characters.

There are three types of labelled records which are:

(a) The plot initial commands;

(b) The plot keyword commands; and

(c) The plot parameter commands.

The plot initial commands are used to define the **plot data** and **boundary** files and to provide identifying information for the plot job output. They are input as one group prior to any other commands. Within the group their order of input is immaterial.

Following the plot initial commands a block of data (labelled or unlabelled records or record images) are required for each plot. The first command of each block is a plot keyward command which instructs the program what type of plot will be be done. The rest of the commands in each block are the plot parameter commands which define the characteristics of the plot. The order of these records within the block does not affect the program execution in any way.

There is no limit on the number of blocks which may be input, and hence, there is no limit on the number of plots which may be plotted. One of the plot keyword commands will signal the beginning of a new plot. The end of the plot(s) is determined by the plot keyword command "**".

The **plot commands** (labelled) are prone to mistakes so that the program provides extensive error checking of all the labelled input. Informative diagnostic messages are printed with the rest of the printer output.

4.4.2 Plot Initial Commands:

There are four plot initial commands *i.e.*, **$DATASET**, **$BNDRYSET**, **$IDENT** and **$TITLE** in this block. Only **$DATASET** is compulsory for a plot job. The **$BNDRYSET** command is conditionally optional on the use of the **$BNDRY** command. (refer to Section 4.4.4(iii)). The order of input of these commands is not important with respect to program execution.

The four commands in this block are described as follows:

(i) The $DATASET Initial Command:

FORMAT (A10, I5)

READ (,) LABEL, NDSET

LABEL – The label is: **$DATASET**

NDSET – The number of **plot data** file(s) residing on the **PLTIN** file.

(ii) The $BNDRYSET Initial Command:

FORMAT (A10, I5)

READ (,) LABEL, NBSET

LABEL – The label is: **$BNDRYSET**

NBSET – The number of **boundary** file(s) residing on the **PLTBND** file.

(iii) The $IDENT initial Command:

FORMAT (A10, 17A4, A2)

READ (,) LABEL, (IDENT(I),I=1,18)

LABEL – The label is: $IDENT

IDENT – Up to 70 characters of identifying information which will be plotted at the beginning of the plot job, *e.g.*, the name and address of the user.

(iv) The $TITLE Initial Command:

FORMAT (A10, I5)

READ (,) LABEL, NLINES

LABEL – The label is: $TITLE

NLINES – 'NLINES' lines of record (number of unlabelled records) needs to be input immediately following the $TITLE initial command. This is required only for the first plot of the job.

4.4.3 Plot Keyword Commands

The plot keyword commands only contain a label. There are four commands, one for each type of plot available and one for program termination.

(a) Label: *VECTOR – signals the beginning of plot parameter input for a *vector* plot, *i.e.*, displacements plots.

(b) Label: *TENSOR – signals the beginning of plot parameter input for a *tensor* plot, *i.e.*, stress trajectories plot.

(c) Label: *PNTCNT – signals the beginning of plot parameter input for a *point contour* plot.

(d) Label ** – signals the end of input and therefore the end of program execution after the last plot has been done.

4.4.4 Plot Parameter Commands

Four plot parameter commands are required to produce any plot. Three of the four commands, **$DATA**, **$BNDRY**, and **$PLOTAREA** , are compulsory input for all plots but the fourth command used depends on the type of plot. For a *vector* or *tensor* plot the **$SCALE** command is necessary. A *point contour* plot requires the **$CONTOURS** command or **$CONTOUR** command; a combination of both can be used. The remaining plot parameter commands are optional.

One feature to remember is that the order of input of plot parameter commands, after a plot keyword command, does not affect program execution.

There are a total of twelve plot parameter commands and they are described individually as follows:

(i) The **$DATA** Command:

READ (,) LABEL, NFDATA,ISLECT

FORMAT (A10, 2I5)

LABEL – The label is: **$DATA**.

NFDATA – The file number of the file on **PLTIN** to be used for the plot.

ISLECT – The selection number used to choose data records from the **plot data** file for plotting. Every ISLECT'th record from the **plot data** file will be selected and used.

(ii) The **$DATAFMT** Command:

READ (,) LABEL, (NDFORM(I), I=1, 18)

FORMAT (A10, 17A4, A2)

LABEL – The label is: **$DATAFMT**

NDFORM – The format for reading records from the **plot data** file specified by NFDATA. The default format code is (1X,5E15.7). Note that the parentheses must be included.

(iii) The $BNDRY Command:

READ (,) LABEL, NFBND,NBPNTS

FORMAT (A10, 2I5)

LABEL – The label is: $BNDRY

NFBND The file number of **boundary** file on **PLTBND** to be used for the plot.

NBPNTS – The value of NBPNTS will be ignored unless NFBND is less than or equal to zero. Otherwise, NBPNTS records (unlabelled records or record images) should follow immediately after the $BNDRY command. These data contain the data records of a temporary boundary information and will be copied onto a temporary file for the duration of the plot. The first record must have NCODE set to 3, otherwise the restrictions the set-up of a **boundary** file on **PLTBND** do not apply here. However, NBPNTS must be greater than or equal to 2.
An 'end-of-file' record must not follow the block of NBPNTS records.

(iv) The $BNDRYFMT Command:

READ (,) LABEL, (NBFORM(I), I=1, 18)

FORMAT (A10, 17A4, A2)

LABEL – The label is: $BNDRYFMT

NBFORM – The format for reading records from the **boundary** file specified by NFBND or the temporary **boundary** file of NBPNTS records associated with the $BNDRY command. The default format code is (1X, 2E15.7, I5). The parentheses must be included.

(v) **The $PLOTAREA Command:**

READ (,) LABEL, XAXIS, YAXIS, XMIN, XMAX, YMIN, YMAX

FORMAT (A10, 6F10.0)

LABEL – The label is: $PLOTAREA

XAXIS – The length of the plotting region, in inches, in the x-direction.
If plotting device is Tektronix the default value is 14.28;
If plotting device is Calcomp the default value is 50.

YAXIS – The length of the plotting region, in inches, in the y-direction.
If plotting device is Tektronix the default value is 10.8;
If plotting device is Calcomp the default value is 33.

XMIN – The minimum global x coordinate of the plotting region.

XMAX – The maximum global x coordinate of the plotting region.

YMIN – The minimum global y coordinate of the plotting region.

YMAX – The maximum global y coordinate of the plotting region.

Note: XMIN, XMAX, YMIN and YMAX are in logical units. Only the data points that lie within the rectangular region bounded by these coordinates will be plotted. This is call the plotting region.
The plotting region also affects the appearance of a boundary taken from a **boundary** file on **PLTBND**. The boundary points on the file and the bounds of the plotting region are combined so that no boundary lines that lies outside the region are used. Intersection points are generated for boundary lines that cross the plotting region bounds. The cut-off ends of boundary loops are connected by appropriate sections of the bounds which are plotted as dashed lines. The composite file produced this method is the boundary plotted for the plot.
Also note that the plotting region does not affects a temporary file input with the $BNDRY command. Care should be exercised to ensure that the boundary points of such a **boundary** file on or lie within the plotting region bounds.

(vi) The $SCALE Command:

READ (,) LABEL, SCALE

FORMAT (A10, F10.0)

LABEL – The label is: $SCALE

SCALE – The data scale factor used in ***vector*** or ***tensor*** plots. It is the number of logical units per plotting inch.

(vii) The $CONTOURS Command:

READ (,) LABEL, CNTINT, MMREAD, ZMIN, ZMAX

FORMAT (A10, F10.0, I10, 2F10.0)

LABEL – The label is: $CONTOURS

CNTINT – The size of the contour interval.

MMREAD – An indicator for using ZMIN and/or ZMAX ,which will be used to define the lower and/or upper bound of contour intervals.
= 0, do not use either ZMIN, ZMAX;
= 1, use ZMIN only;
= 2, use ZMAX only; and
= 3, use both ZMIN and ZMAX.
The other alternative is to use the maximum and minimum values of Z (the data to be plotted) found on the data file.

Note: This command is used to define a range of contour intervals of equal size. Only one may be input for a plot. ZMIN and ZMAX, or their alternatives, whichever are used, are adjusted down or up, respectively, the minimum amount required to make them multiples of CNTINT.

(viii) The $CONTOUR Command:

READ (,) LABEL, CMIN,CMAX

FORMAT (A10, 2F10.0)

LABEL – The label is: $CONTOUR

CMIN – The lower bound of the contour interval.

CMAX – The upper bound of the contour interval.

Note: This command is used to define a single contour interval. As many as twenty **(20)** of this command may be input for a plot job.

(ix) The $LABEL Command:

READ (,) LABEL, NOLABL, LABELS

FORMAT (A10, I5, /20A4)

LABEL – The label is: $LABEL

NOLABL – The number of labels (records or record images) to be input immediately following the $LABEL command.

LABELS – 80 characters (unlabeled) can be used for each label. A maximum of five labels (records) for an individual plot is permitted.

(x) The $RANGE Command:

READ (,) LABEL, RMIN, RMAX

FORMAT (A10, 2F10.0)

LABEL – The label is: $RANGE

RMIN – The minimum data value to be plotted.

RMAX – The maximum value to be plotted.

Note: RMIN and RMAX define a range of data values to be used in a *point contour* plot. Only points having a Z value within this range will be plotted.

(xi) The $POLAR Command:

READ (,) LABEL

FORMAT (A10)

LABEL – The label is: $POLAR.
This command signals the program that the point coordinates on the **plot data** file are in polar form rather than in cartesian form. Note that all angles must be in degrees.

(xii) The $NOSPACE Command:

READ (,) LABEL

FORMAT (A10)

LABEL – The label is: $NOSPACE.

Note: This command is a special option that can only be used with *vector* and *tensor* plots. It signals the program to draw this plot over the previous one, using the same plotting region. This allows the superposition of two or more plot input one after another. When this command is used, the $DATA, $DATAFMT,$LABEL and $POLAR are the only other commands allowed as input for a plot.

4.4.5 Error Checking

All labelled commands are checked for errors as they are input. Most of the types of errors detected can be summarized by the following descriptions.

(a) Unknown command;

(b) Misplaced input - input with the wrong section;

(c) Repeated input - input more than once;

(d) Missing input - certain data must be input; and

(e) Improper parameter - values out of range.

All of the errors are classified into three levels of severity. The particular level of an error depends on which command (labelled record) is involved. Each level initiates a distinct action in the program. They are listed below in order of increasing severity.

Each is accompanied by a short description of the action taken by the program as a direct result of the error.

(a) WARNING	–	record ignored; input next record.
(b) *ERROR*	–	This plot will not be attempted; proceed with input for the next plot.
(c) **FATAL ERROR**	–	Program terminates execution after closing the plot file.

4.5 Interactive SYS$COMMAND Data Input

There are two separate interactive SYS$COMMAND data input groups:

(a) The **plot legend** parameters data; and

(b) The **pause** command.

The **plot legend** parameters data are needed for plotting the scale legend on the plots. There are total of sixteen **16 plot legend** parameters to be interactively input when the ***MRLPLT*** is executing.

The **pause** (Y/N) command are used to verify the **plot legend** parameters data during the data input and to activate and set Calcomp 1042 plot origin.

4.5.1 **Plot Legend** Parameters

(i) READ (,) MUNIT

FORMAT (A3)

MUNIT – (1-3) character string that identifies the unit of the model.

(ii) READ (,) DUNIT

FORMAT (A3)

DUNIT – (1-3) character string that identifies the unit of the displacements.

(iii) READ (,) SUNIT

FORMAT (A3)

SUNIT – (1-3) character string that identifies the unit of the stresses.

(iv) READ (,) PLTDEV

FORMAT (A3)

PLTDEV – 3 character string that identifies the plot device.
Available devices are:
TEK - Tektronix PLOT10 graphic display terminal.
CAL - Calcomp 1042 plotter.

(v) READ (,*) ITYPE

ITYPE – (1–3) to identify the three different MRLPLT plots.
ITYPE = 1 vectors plot;
ITYPE = 2 tensors plot;
ITYPE = 3 point contours plot.
* *Note*: Denotes free format.

(vi) READ (,) MPIND

FORMAT (A1)

MPIND – 1 character string (Y/N) to indicate whether the material properties is to be plotted on the plot legend.

If MPIND = "Y"

(vii) READ (,*) NUMMAT

NUMMAT – Number of material properties to be plotted on plot legend.

(viii) READ (,*) NSF

NSF – (1–14) to identify type of point contours plot to perform.
NSF = 1 Mohr–Coulomb criterion;
NSF = 2 Maximum shear (τ_f / τ_{max}) approach;
NSF = 3 Drucker/Prager criterion;
NSF = 4 Hoek/Brown strength/stress ratio;
NSF = 5 Major principal stress;
NSF = 6 Minor principal stress;
NSF = 7 Intermediate principal stress;
NSF = 8 σ_x;
NSF = 9 σ_y;
NSF = 10 σ_z;

NSF = 11 Major principal strain;
NSF = 12 Minor principal strain;
NSF = 13 Intermediate principal strain;
NSF = 14 Volume dilatation.

If MPIND = "Y" and NSF = 1 or 2 or 3

(ix) READ (,*) COHEC

COHEC – Cohesive strength of rock material.

(x) READ (,*) PHI

PHI – Internal friction angle of rock material.

If MPIND = "Y" and NSF = 4

(xi) READ (,*) XM

XM – Hoek/Brown empirical constant m.

(xii) READ (,*) XS

XS – Hoek/Brown empirical constant s.

(xiii) READ (,*) QU

QU – Uniaxial compressive strength of intact rock.

If MPIND = "Y" and NSF = 11 or 12 or 13 or 14

(xiv) READ (,*) E

E – Youngs modulus of rock material.

(xv) READ (,*) XNU

XNU – Poisson's ratio of rock material.

(xvi) READ (,*) THIT

THIT – Character height of the plot legend.

(xvii) READ (,*) CORVAL

CORVAL – Co-ordinate scale factor to be shown on the **plot legend.**

(xviii) READ (,*) VECVAL

VECVAL – Scale factor for *vector* plot to be shown on the **plot legend .**

(xix) READ (,*) TENVAL

TENVAL – Scale factor for *tensor* plot to be shown on the **plot legend .**

(xx) READ (,*) YPAGIN

YPAGIN – Plot width in inches (y-direction).
Note: YPAGIN is used only when YAXIS on $PLOTAREA is 0.

(xxi) READ (,*) XPAGIN

XPAGIN – Plot length in inches (x-direction).
Note: XPAGIN is used only when XAXIS on $PLOTAREA is 0.

5.0 OUTPUT

5.1 Printer Output

The printer output consists almost entirely of the reproduction of all command input (labelled records) presented in a more understandable format. This includes all of the information output on the plotter except for the data points. Any heading lines cut off on the plotter output due to paper width are given in full in the printer output. The actual boundary points used for each plot boundary are also listed. A message preceding the list indicates whether the boundary was obtained from input (with the

$BNDRY command) or produced from a **boundary** file on **PLTBND**. For a *point contour* plot the legend is reproduced with all numerical values given to four significant digits. In addition, all diagnostic messages are displayed with the sections of the plot job where the error conditions were encountered.

5.2 Plotter Output

Each plot may appear a maximum of two 70 characters line of plot information as heading, a set of boundary lines, and, within the plotting region, the symbols representing the data point values. The border, preserved on all sides of the plotting region, is blank except for the symbols of the *vector* or *tensor* plots which may extend into it.

The vector symbol is an arrow, with a very small arrowhead at the terminal point, drawn perpendicular to a short base line at the initial point. If the terminal point of a vector lies outside the borders the arrow is terminated at the border edge and no arrowhead is drawn. Due to pen resolution any vector whose scaled magnitude is less than 0.03 inches will be drawn as a small cross (+), 0.03 inches high, at the initial point. A message to this effect is plotted along with the vector scale factor, input with $SCALE command, in the heading of the plot following the coordinate scale factor.

The symbol used in a *tensor* plot is two line segments, mutually perpendicular, bisecting each other at the initial point. The lengths of the line segments represent twice the magnitude of the principal values, *e.g.*, the major and minor principal stresses. The line segments are drawn so that the major principal value (stress) is inclined at its specified angle, *i.e.*, the direction of the principal stresses. Negative values (compressive stresses) are distinguished by a short bar drawn perpendicular to the line segment at each end.

The contour intervals of a *point contour* plot are represented by alphanumeric characters. In ascending numerical order the sequence of symbols is A to Z followed by 0 (zero) to 9 for a maximum of 36 closed intervals. Plus (+) and minus (-) signs are used to indicate the open intervals above the last interval and below the first interval, respectively. Each data point within the plotting region is assigned to a contour interval according to its value. The appropriate symbol is plotted such that the initial point is the bottom left corner of the symbol. A complete legend, giving numerical values to two significant digits, is plotted beside the plotting region. The use of the $RANGE command is not indicated in the legend. However, no symbols are plotted for the data points whose values lie outside the specified range.

5.3 Plotting Examples

An example plot job has been devised to show the general nature of the input stream and to illustrate the type of plot output to be expected from *MRLPLT*. The data for these plots was taken from the results of a finite element analysis using the *SAP2D* program.

The input data, as shown in Table 1, is an example of the ***MRLPLT*** **plot command file.** Table 2 shows an example of the **boundary** file, which is generated by the boundary generator ***BNDGEN*** [5]. The **plot data** file , generated by the interface program ***SAPPLT*** [5], is too large to be listed here. Note that the files **PLTBND (boundary** input data file),and **PLTCMD (plot commands** file) and **PLTIN (plot data** file) were created by ***SAPPLT*** and ***BNDGEN***, respectively, before the plot program was run.

Three plots, as shown in figures 1, 2 and 3, illustrate the features of the three different types of plots.

Table 1. An Example of ***MRLPLT*** **Plot Commands** Data File

```
$DATASET       3
$BNDRYSET      1
$IDENT     MY NAME
$TITLE         2
EXAMPLE - TWO 5X14 M ROOMS SEPARATED BY A 5X10 M PILLAR
          DUE TO SYMMETRY ONLY ONE HALF OF THE STRUCTURE IS ANALYZED.
*TENSOR
$DATA          1     1
$LABEL         1
STRESS TRAJECTORIES
$BNDRY         1     0
$PLOTAREA        7.0        7.0        0.0       25.0       90.0      110.0
$SCALE         100.0
*VECTOR
$DATA          2     1
$LABEL         1
DIFFERENTIAL DISPLACEMENT
$BNDRY         1     0
$PLOTAREA        7.0        7.0        0.0       25.0       90.0      110.0
$SCALE          0.001
*PNTCNT
$DATA          3     1
$LABEL         1
MAJOR PRINCIPAL STRESSES
$BNDRY         1     0
$PLOTAREA        7.0        7.0        0.0       25.0       90.0      110.0
$CONTOURS        1.0          0
**
```

Table 2. An Example of The **Boundary** Data File

0.0000000E+00	0.0000000E+00	3
0.1000000E+03	0.0000000E+00	2
0.1000000E+03	0.2050000E+03	2
0.0000000E+00	0.2050000E+03	2
0.0000000E+00	0.0000000E+00	2
0.5000000E+01	0.1000000E+03	3
0.5000000E+01	0.1050000E+03	2
0.1900000E+02	0.1050000E+03	2
0.1900000E+02	0.1000000E+03	2
0.5000000E+01	0.1000000E+03	2

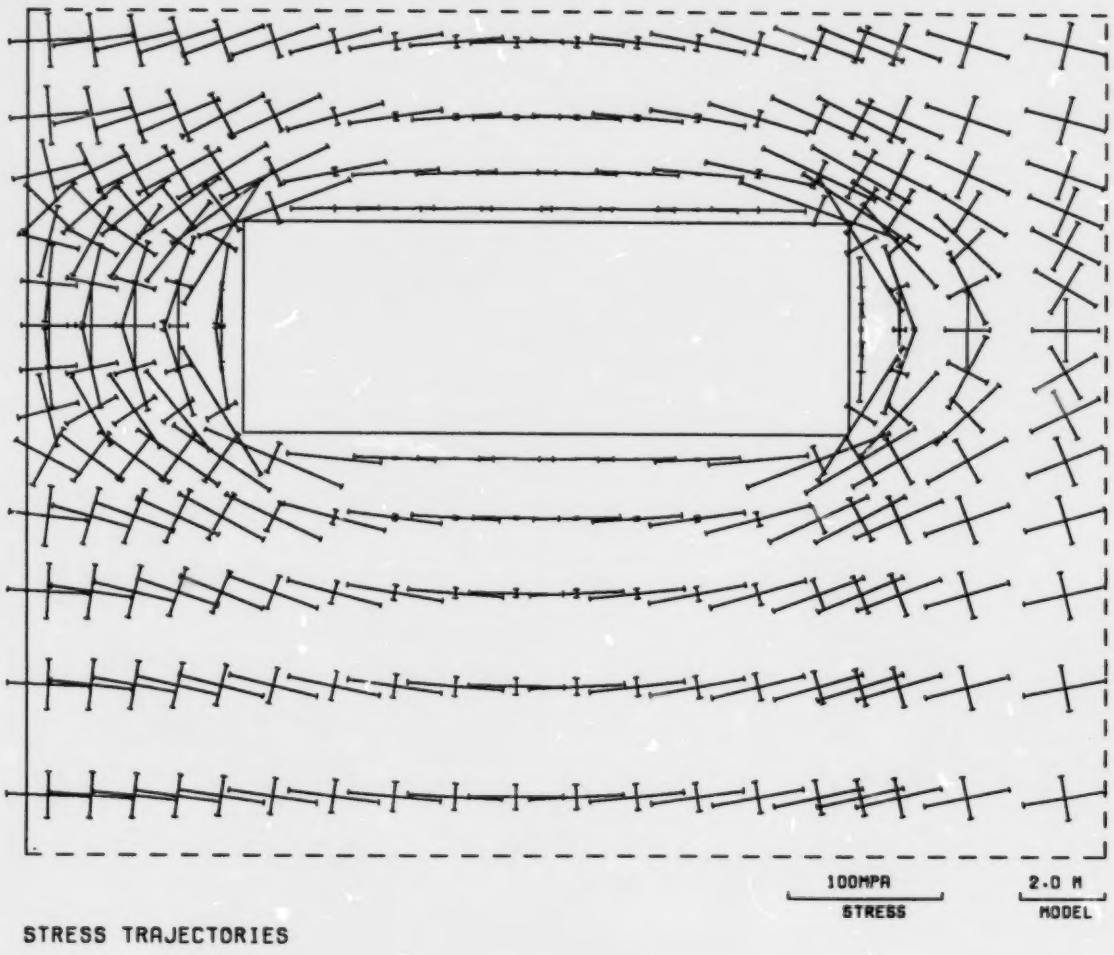

Figure 1. *Tensor* Plot - Stress Trajectories

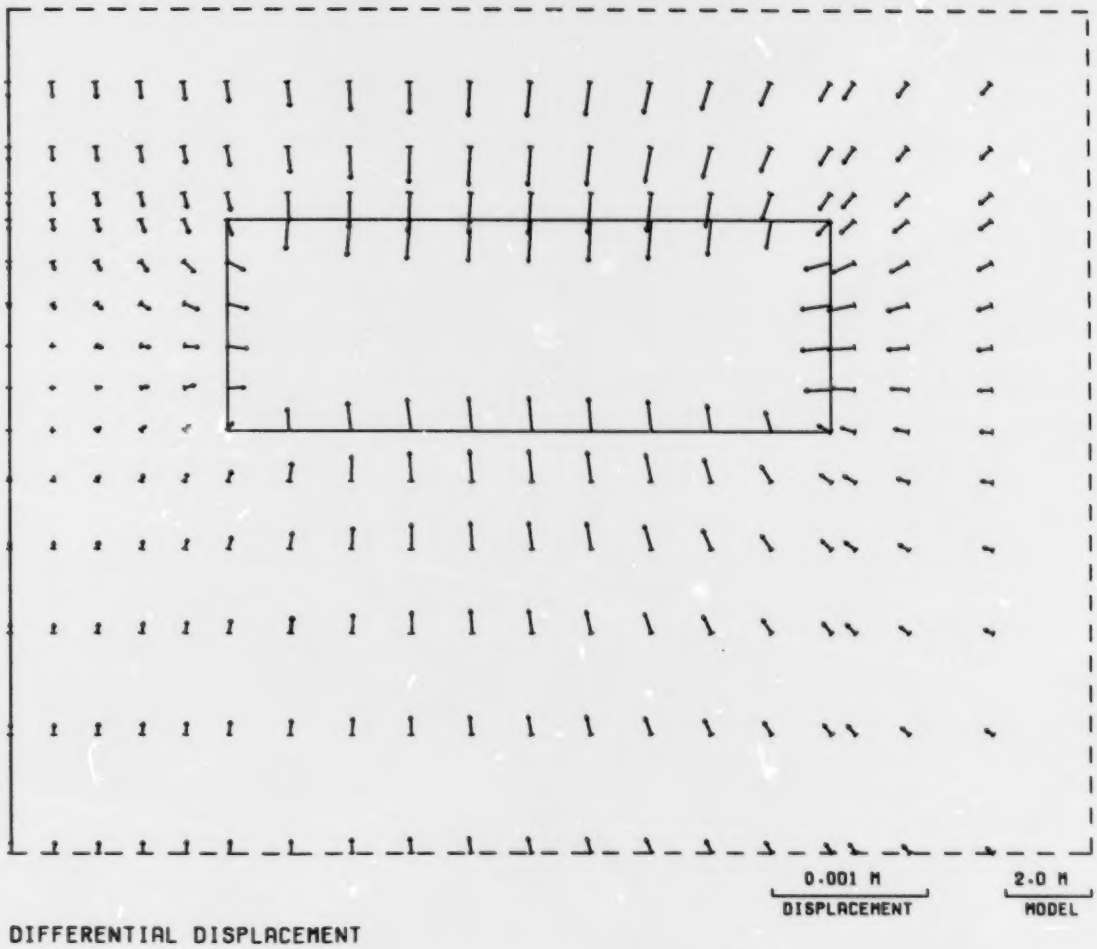

Figure 2. *Vector* Plot - Excavation Displacements

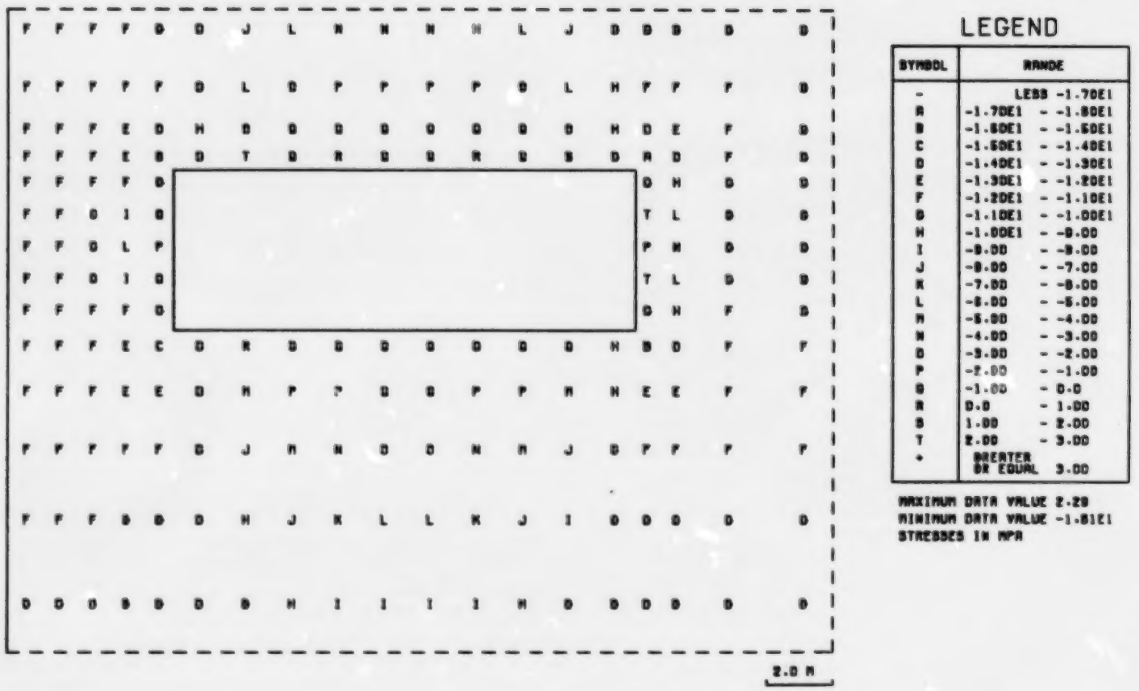

Figure 3. *Point Contour* Plot - Major Principal stresses

6.0 *MRLPLT* COMMAND PROCEDURE

To facilitate the use of *MRLPLT*, a command procedure has been written. It prompts the user the names of input **plot data** file, **boundary** data file, **plot commands** data file and, gives the user various options for the output plotting files produced. The command procedure is listed in Appendix A.

The command procedure does the following:

i) The user is prompted for the names of the input **boundary** data file. The procedure checks that the files exist.

ii) The user is prompted for the names of the input **plot commands** data file. The procedure checks that the files exist.

iii) The user is prompted for the names of the input **plot data** file. The procedure checks that the files exist.

iv) The user is prompted for the names of the plotting device, *i.e.*, "TEK" or "CAL".

v) The user is prompted for the name of the file to hold the plotting file.

vi) If "TEK" is selected, then the Tektronix version of *MRLPLT* is executed. The **plot legend** parameters are input during the *MRLPLT* execution.

vii) If "CAL" is selected, then the Calcomp version of *MRLPLT* is executed. The **plot legend** parameters are input and the user is prompted to answer various **pause** commands during the *MRLPLT* execution.

7.0 REFERENCES

1. Morrison, H. D.; "Documentation of a Plot Program"; Report MRP/MRL 75-34 (TR); Mining Research Laboratories, CANMET, Energy, Mines and Resources Canada; Ottawa; April, 1975.

2. Toews, N. A., and Wong, A. S.; "Documentation Of TEKIMP"; Division Report M&ET/MRL 86-9 (INT); Mining Research Laboratories, CANMET, Energy, Mines and Resources Canada; Ottawa; January, 1986.

3. Toews, N. A., and Wong, A. S.; "Documentation Of TEKPS"; Division Report M&ET/MRL 87-54 (INT); Mining Research Laboratories, CANMET, Energy, Mines and Resources Canada; Ottawa; January, 1989.

4. Toews, N. A., Yu, Y. S., and Wong, A. S.; "SAP2D Documentation (VAX -11/750 Version) - A 2-D Linear Elastic Finite Element Computer System"; Division Report MRP/MRL 85-35 (TR); Mining Research Laboratories, CANMET, Energy Mines and Resources Canada; Ottawa; August, 1985.

5. Yu, Y. S., Toews, N. A. and Wong A. S.; "SAPPLT Documentation (VAX -11/750 Version) - An Interface Program for the General Purpose Plotting Program MRLPLT"; Division Report MRP/MRL 85-86 (TR); Mining Research Laboratories, CANMET,Energy, Mines and Resources Canada; Ottawa; August, 1985.

6. Toews, N. A., Yu, Y. S., Vongpaisal, S. and Wong, A. S.; "Documentation of DRUKPLT"; Division Report MRP/MRL 84-81(INT); Mining Research Laboratories, CANMET, Energy, Mines and Resources Canada; Ottawa; September 1984.

APPENDIX A – *MRLPLT* COMMAND PROCEDURE

```
$ ! *** MRLPLT COMMAND PROCEDURE ***
$ !
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "YOU ARE RUNNING THE MRLPLT COMMAND PROCEDURE"
$ !
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "THE INPUT FILES REQUIRED ARE:"
$ WRITE SYS$OUTPUT "    1) PLOT BOUNDARY DATA   - PLTBND"
$ WRITE SYS$OUTPUT "    2) MRLPLT COMMAND DATA  - PLTCMD"
$ WRITE SYS$OUTPUT "    3) MRLPLT PLOT DATA     - PLTIN"
$ !
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "THE PLOT FILE PRODUCED IS:"
$ WRITE SYS$OUTPUT "    1) MRLPLT PLOT FILE "
$ !
$AGAINA1:
$ !
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "INPUT MRLPLT BOUNDARY DATA FILE"
$ INQUIRE PBNDIN "DEFAULT - PLTBND.DAT -"
$ IF PBNDIN.EQS."" THEN PBNDIN :="PLTBND.DAT"
$ PBNDIN = F$SEARCH(PBNDIN)
$ IF PBNDIN.NES."" THEN GOTO AGAINA2
$ WRITE SYS$OUTPUT "SPECIFIED FILE DOES NOT EXIST"
$ GOTO AGAINA1
$ !
$AGAINA2:
$ !
$ ASSIGN 'PBNDIN' PLTBND
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "INPUT MRLPLT COMMAND FILE"
$ INQUIRE PCMDIN "DEFAULT - MRLPLTCMD.DAT -"
$ IF PCMDIN.EQS."" THEN PCMDIN := "MRLPLTCMD.DAT"
$ PCMDIN = F$SEARCH(PCMDIN)
$ IF PCMDIN.NES."" THEN GOTO AGAINA3
$ WRITE SYS$OUTPUT "SPECIFIED FILE DOES NOT EXIST"
$ GOTO AGAINA2
$ !
$AGAINA3:
$ !
$ ASSIGN 'PCMDIN' PLTCMD
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT "INPUT MRLPLT PLOT INPUT DATA FILE"
$ INQUIRE PIN "DEFAULT - MRLPLTDAT.DAT -"
$ IF PIN.EQS."" THEN PIN := "MRLPLTDAT.DAT"
$ PIN = F$SEARCH(PIN)
$ IF PIN.NES."" THEN GOTO AGAINA4
$ WRITE SYS$OUTPUT "SPECIFIED FILE DOES NOT EXIST"
$ GOTO AGAINA3
```

```
$ !
$AGAINA4:
$ !
$ ASSIGN 'PIN' PLTIN
$ !
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "INPUT PLOTTING DEVICE :-"
$ WRITE SYS$OUTPUT "  TEK = TEKTRONIX 4014; CAL = CALCOMP 1042"
$ INQUIRE PLTDEV "  DEFAULT - TEK"
$ IF PLTDEV.EQS."CAL" THEN GOTO NEXTA1
$ !
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "INPUT MRLPLT PLOT FILE"
$ INQUIRE PLOT "DEFAULT - PLOTFILE.TEK -"
$ IF PLOT.EQS."" THEN PLOT := "PLOTFILE.TEK"
$ ASSIGN 'PLOT' TEKOUT
$ !
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "PERFORM TEKTRONIX 4014 MRLPLT PLOT"
$ WRITE SYS$OUTPUT " "
$ ASSIGN SYS$COMMAND FOR005
$ RUN DISK$USER:[FE.COMMAND]MRLPLTTEK
$ !
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "TEKTRONIX 4014 MRLPLT PLOT COMPLETED"
$ WRITE SYS$OUTPUT " "
$ !
$ GOTO NEXT
$ !
$NEXTA1:
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "PERFORM CALCOMP MRLPLT PLOT"
$ WRITE SYS$OUTPUT " "
$ ASSIGN SYS$COMMAND FOR005
$ RUN DISK$USER:[FE.COMMAND]MRLPLTCAL
$ !
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "CALCOMP MRLPLT PLOT COMPLETED"
$ WRITE SYS$OUTPUT " "
$ !
$NEXT:
$ !
$FINISH:
```